

The Globus Galaxies Platform: Delivering Science Gateways as a Service

Ravi Madduri¹, Kyle Chard¹, Ryan Chard², Lukasz Lacinski¹, Alex Rodriguez¹,
Dinanath Sulakhe¹, David Kelly¹, Utpal Dave¹, Ian Foster¹

1. Computation Institute, University of Chicago and Argonne National Laboratory, Chicago, USA
2. School of Engineering and Computer Science, Victoria University of Wellington, New Zealand

Abstract

The use of public cloud computers to host sophisticated scientific data and software is transforming scientific practice by enabling broad access to capabilities previously available only to the few. The primary obstacle to more widespread use of public clouds to host scientific software (“cloud-based science gateways”) has thus far been the considerable gap between the specialized needs of science applications and the capabilities provided by cloud infrastructures. We describe here a domain-independent, cloud-based science gateway platform, the Globus Galaxies platform, which overcomes this gap by providing a set of hosted services that directly address the needs of science gateway developers. The design and implementation of this platform leverages our several years experience with Globus Genomics, a cloud-based science gateway that has served more than 200 genomics researchers across 30 institutions. Building on that foundation, we have implemented a platform that leverages the popular Galaxy system for application hosting and workflow execution; Globus services for data transfer, user and group management, and authentication; and a cost-aware elastic provisioning model specialized for public cloud resources. We describe here the capabilities and architecture of this platform, present six scientific domains in which we have successfully applied it, report on user experiences, and analyze the economics of our deployments.

1. Introduction

Science as a service[1]—making scientific data and software available through Internet-accessible services—is now a common approach for developing scientific software. Scientists and gateway developers alike have embraced Software as a Service (SaaS) approaches due to the benefits of operating a single copy of software and thus lowering development and support costs. Software consumers have also benefited from increased access to reliable and supported software versions, without needing to install, operate, or even understand the inner workings of the software.

Notable examples of successful scientific service providers include Agave[2], Globus[3], HUBzero[4], and Apache Airvata[5]. However, while these efforts provide useful scientific capabilities via service interfaces, they focus on low-level building block capabilities rather than domain-specific end-to-end capabilities. Conversely, science gateways focus on end-to-end solutions for a single domain and are therefore limited in terms of the research communities that can be served. Approaches such as

Apache Airavata (now the basis of the Science Gateways Platform as a Service project) provide a suite of important services that can be used by gateways to outsource specific tasks. However, considerable work is still required to develop a gateway using these building blocks. While some gateways such as HUBzero have been generalized, adapted, and applied to other domains, the aim for generality remains rare. Most often gateways are developed with a specific, vertically integrated software stack, dependent, for example, on the underlying resources used for execution and the specific identity providers employed. Thus, adaption to other environments and domains is challenging.

Cloud computing is transforming scientific practice by providing access to huge computing capacity, support for scaling elastically to meet fluctuating demands, and charging users only for the resources used. As such, there has been a rapid growth in the number of cloud-based science gateways developed across a wide range of scientific domains[6]. However, each gateway represents a significant development effort as there is still a considerable gap between the needs of scientific applications and the capabilities provided by cloud infrastructure. In many cases, research groups who would benefit from cloud-based science gateways do not have the technical or operational capabilities available to develop, deploy and operate them. Therefore, most gateway success stories are associated with large and well-funded user communities. We believe that the same benefits that large communities gain from science gateways, such as providing abstracted and easy to use access to complex and computationally intensive analyses and large shared datasets, can be transformative to the research practices of smaller research groups.

Science gateways are rapidly evolving. No longer do they focus primarily on simplifying access to large-scale high performance computing (HPC) resources. Instead, they provide complete collaboration suites for scientists. Thus, gateway developers are faced with challenges supporting complex collaborative interactions with respect to analysis and data management. In isolation, these issues are not prohibitive, however they represent additional development and operational costs. We therefore believe that new SaaS-based approaches are required to lower operational costs and develop a path towards long-term sustainability.

We present here the Globus Galaxies platform, a cloud-hosted solution to the problem of deploying and operating science gateways. The platform is based on our previous three years of experience developing and operating Globus Genomics [7], a software as a service (SaaS) provider of analysis capabilities for

the genomics community. The resulting Globus Galaxies platform is flexible and can be used to easily enable SaaS delivery of scientific analyses and data management capabilities for specific domains (i.e., a science gateway). A flexible workflow-based interface allows users to create arbitrarily complex analyses using a standardized and extensible model for including new tools and software. A complete data management solution allows researchers to easily access, share, and transfer huge datasets and to leverage commercial infrastructure as a service (IaaS) providers to elastically scale analysis execution. A carefully designed, modular architecture makes it straightforward to create new deployments customized to meet the unique requirements of new scientific domains.

The rest of this paper is organized as follows. In Section 2 we describe the Globus Galaxies platform including details about its deployment model and cost-aware scheduling capabilities. In Section 3 we discuss unique challenges associated with using cloud resources for scientific analyses. We describe six scientific gateways built upon the platform in Section 4. In Section 5 we evaluate the usage of the platform and the benefits obtained from scheduling workloads with cost-aware scheduling approaches. In Section 6 we discuss related work and finally in Section 7 we provide a summary of the paper.

2. The Globus Galaxies platform

The Globus Galaxies platform is designed to enable the rapid and cost-effective creation and operation of cloud-hosted SaaS scientific gateways. The platform is built upon powerful capabilities provided by the Galaxy workflow system [8], for the flexible creation and execution of scientific workflows; Globus transfer [9], for easy, reliable, and secure access, sharing and transfer of large amounts of data; Globus Nexus [10] for identity management and group-based access control; Swift [11] for parallel execution of workflow components; and other components, such as HTCondor [12], for efficient scheduling. The platform is designed to elastically scale by dynamically provisioning on-demand cloud computing resources [13]—all without exposing end users to the complexities of managing advanced software and large-scale infrastructure. Its high level architecture is shown in Figure 1.

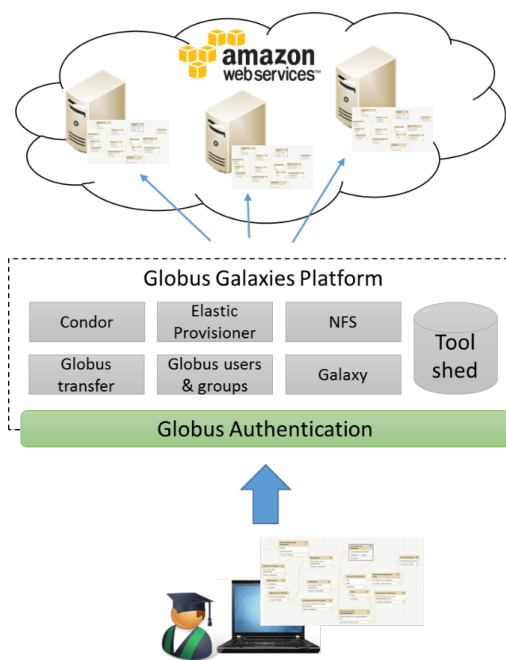


Figure 1: Globus Galaxies platform architecture.

2.1 Galaxy

Many scientific analyses involve sequential and iterative processing using a variety of tools. The Globus Galaxies platform leverages the Galaxy workflow engine to provide these core analysis capabilities. Galaxy provides a web-based interface for creating, executing, sharing, and reusing workflows composed of arbitrary applications, tools, and scripts. Galaxy records the analysis steps performed by a user. At the completion of an analysis the user may save these steps as a workflow that can be re-executed or shared with others. Galaxy was initially developed for large-scale data-oriented biomedical analyses, and has been used by thousands of biomedical users who have demonstrated its ability to handle large-scale analyses.

Galaxy’s extensible plug-in architecture allows users to customize data types and tools for any domain. Before being used in Galaxy, tools must be “wrapped” with XML files describing their input and output data types, executable location, and invocation arguments. Galaxy uses this information to present customized user interfaces with specific input parameters and to link the output of one tool to the input of another. Galaxy enforces type checking by providing visual feedback to users when connecting tools that have incompatible inputs and outputs. Galaxy provides a tool repository (called a “tool shed”) for storing and sharing tool wrappers, data types and exported workflows. A global tool shed operated by the Galaxy team provides thousands of freely available (biomedical) tools that can be found and installed on any Galaxy deployment.

While Galaxy is developed as a web application, it is primarily designed to be installed and managed locally. The Galaxy team operates a public Galaxy instance that is freely available to users, but this service is shared by many thousands of people and thus cannot provide high quality of service (QoS) guarantees. Instead, the shared instance allows users to experiment with the system before they install it locally for large-scale use. Furthermore, while the public Galaxy instance is equipped with many biomedical tools, it cannot be extended with other tools or to support other data types and disciplines.

2.2 Workflow execution

Once a user has loaded or developed a workflow, Galaxy provides mechanisms for executing this workflow on local and remote resources. While Galaxy is primarily designed to use local compute resources co-located with the Galaxy server, it can also support cluster resources through its extensible “runner” interfaces.

The Globus Galaxies platform builds upon Galaxy’s runner interfaces to support a dynamically provisioned cluster using the HTCondor scheduler. The HTCondor master node and queue are hosted on the Globus Galaxies instance, which also hosts Galaxy. HTCondor worker nodes are dynamically connected to the HTCondor master node when a cloud instance is instantiated via contextualization scripts. In order to ensure that shared data is available to any worker node a network File System (NFS) hosted by the Globus Galaxies instance is used.

Galaxy focuses on parallelizing execution of entire workflows. However, in many scientific use cases, the components of the workflow themselves are implicitly parallel (e.g., loops). There are significant challenges associated with enabling efficient parallel execution of sequential tools. We employ the Swift system in the Globus Galaxies platform as a method for wrapping tools and enabling parallel execution. Swift is a parallel scripting language designed for composing applications into workflows that can be executed in parallel on multicore processors, clusters, grids, clouds and supercomputers; it can also be used to parallelize individual applications. Users adding tools to a Globus Galaxies instance may choose to wrap their tool in a Swift wrapper, in addition to the Galaxy wrapper required for use by Galaxy. The Swift wrapper parallelizes invocation of the tool based on the arguments supplied when executed, and thus enables large-scale parallel execution. Internal tool and sub-tool executions within the Swift wrapper are transparent to Galaxy and users.

2.3 PaaS integration

The Globus Galaxies platform leverages external cloud services for high performance data, user and group management.

2.3.1 Globus Nexus for identities and groups

Globus Nexus provides identity, profile, and group management as a service. Its identity management capabilities allow users to create a Globus identity that can be used for Single Sign-On (SSO) across services that also leverage Globus identities. Globus Nexus’s group management capabilities permit users to create and manage their own groups, assign roles to group members, and use groups for authorization.

The Globus Galaxies platform integrates with Globus Nexus to outsource all identity and group management tasks. Thus, Globus Galaxies does not manage users itself, nor does it need to provide sophisticated identity management workflows for such tasks as validating email addresses or resetting forgotten passwords. It uses Globus identities across the platform; for example enforcing data transfer access restrictions based on this same identity. Globus Galaxies uses groups managed in Globus Nexus for authorization and access to shared instances.

2.3.2 Globus Nexus for authentication

Globus Nexus acts as an identity hub for associating external identities with a Globus Identity, thereby allowing users to authenticate using different authentication protocols. By integrating these capabilities users of the Globus Galaxies platform can authenticate using their Globus identity or any linked

identity, for example their campus identity via CILogon/InCommon, their Google account via OpenID, or their Extreme Science and Engineering Discovery Environment (XSEDE) account using MyProxy OAuth.

These capabilities are integrated via an external authentication module within the Web Service Gateway Interface (WSGI) middleware used in Galaxy. User authentication follows the typical three-legged OAuth-2 authentication workflow. When a user attempts to log in, the page redirects the user to authenticate using Globus Nexus. Once the user successfully authenticates (using any of their linked external identities), Globus OAuth redirects the user back to the Globus Galaxies instance with a short-lived access token. This access token is then verified and stored in an in-memory database that stores the mapping between the Galaxy session and the Globus identity and OAuth token.

2.3.3 Globus transfer for reliable data movement

Globus transfer provides high performance, secure, third party data movement, synchronization and sharing. It automates challenging aspects of the data transfer process, tuning parameters to maximize bandwidth, managing security configurations, providing automatic fault recovery, and notifying users of errors and completion. Its sharing capabilities allow users to manage access to data stored in remote storage services.

The Globus Galaxies platform leverages these capabilities to support automated movement of large amounts of data between remote endpoints (e.g., sequencing centers, image acquisition machines, institutional storage, and PCs) to and from the Globus Galaxies platform. It allows users to select a data source from which to download data (e.g., an acquisition machine) and/or to select destination endpoints to which data is deposited. Integration with Globus Nexus provides the support for authentication operations to ensure that access to data is permitted.

2.4 Elastic cloud execution

Many scientific fields face “Big Data” challenges, requiring huge computing resources to derive results from large quantities of data. Analyzing a single genome or computing a single climate model over a small area may take several days on a single processor. Many researchers now wish to analyze several hundred genomes or to look at the interactions between multiple climate models over large geographical regions. Unfortunately, few researchers have access to large-scale compute resources, and purchasing such resources is often cost prohibitive.

The Globus Galaxies platform is designed to make use of commercial cloud resources to provide flexible and cost-efficient execution of analyses. Cloud computing is particularly attractive in this scenario as users pay only for the resources they use, and face no upfront investment costs. Clouds provide efficient execution, on-demand provisioning of compute resources when required, and the release of those resources when they are not in use. Analyses can be elastically scaled horizontally with little overhead, for example, 1000 genomes can be analyzed in the same amount of time as a single genome by provisioning 1000 instances. Similarly, the cost of analyzing 1000 genomes in this fashion is the same as analyzing 1000 genomes in sequence on the same instance. Support for virtual machines is also important as it allows customized execution environments to be created with specific software versions, dependencies and datasets.

The Globus Galaxies platform has been designed to operate on Amazon Web Services Elastic Compute Cloud (EC2); however it could be extended to use other cloud platforms, for example we

have applied the platform on the Magellan Cloud at Argonne National Laboratory.

The Globus Galaxies platform relies on a custom *elastic provisioner* that provisions cloud instances on demand using cloud computing APIs. The elastic provisioner monitors the Globus Galaxies HTCondor job queue to determine real-time job wait times. If the queue length or wait time exceeds a predefined limit, the scheduler creates new HTCondor worker nodes on cloud instances and associates these nodes with the HTCondor scheduler.

Globus Galaxies supports two approaches to configuring new worker node instances: using a predefined AMI or dynamically contextualizing a new instance. In the first approach we rely on a pre-created AMI that includes all Globus Galaxies software as well as any tools that are included in the specific instance. The elastic provisioner uses the cloud-init utility to contextualize this AMI so that it is configured to interact with the appropriate Globus Galaxies head node (e.g., HTCondor master and NFS server). Cloud-init is a widely used script-based system for installing and configuring software on a newly created instance. In the second approach, the elastic provisioner uses cloud-init to completely contextualize a bare-bone AMI from scratch, loading all required software and workflow components when it is created. In situations with few software dependencies, this approach provides flexibility with little overhead, but if many software packages and dependencies must be installed, contextualization overhead can be considerable.

2.5 Cost aware scheduling

Given the underlying utility computing models employed by commercial cloud providers, in which users pay for resources used, there is potential for a high degree of economic inefficiency if instances are provisioned naively. To address this issue, the elastic provisioner includes cost-aware scheduling functionality to select instances based on real-time economic conditions.

Amazon EC2, like many other commercial cloud providers, has several pricing models that can be used to reserve instances. The two models used by Globus Galaxies are: on demand and spot pricing. On demand instances are advertised following a static posted price model, in which the requesting user knows the price (per hour) of a particular instance in advance. Spot pricing is a dynamic model in which instances are charged at the current market price. Users bid a maximum price at which they are willing to pay, if at some later time the spot price increases to the point that it exceeds the original bid, the instance is reclaimed. Spot prices provide significant cost savings over on demand instances, however they do so at the expense of reliability.

Given the potential for a wide range of scientific analysis types and their associated usage requirements and QoS constraints the Globus Galaxies platform supports provisioning via on demand or spot prices. Most often, worker nodes are hosted on spot instances. The maximum bid price is typically determined by the elastic provisioner based on a pre-defined policy defined for the particular Globus Galaxies instance (e.g., \$4 per hour for c3.4xlarge). If the spot price exceeds the maximum bid price, and the instance is terminated, HTCondor will return any jobs that were running on the terminated instance to the queue until the provisioner is able to provision new resources. Any previous execution state is lost. If particular jobs are marked as important or have strict deadlines, the elastic provisioner may use on-demand instances.

To further improve cost efficiency the provisioner is configured with a list of instance types that can satisfy the requirements of jobs for a particular Globus Galaxies instance. The provisioner compares spot prices across availability zones for each instance type and bids for the cheapest instance type at that point in time that satisfies or exceeds the computational capability of the application that is being executed. This approach alleviates the issue of our own Globus Galaxies worker instances influencing the spot price. Which can occur when many spot requests are made for the same instance type in the same availability zone by different Globus Galaxies instances. It also lowers wait times and costs, as we expand our bidding to multiple instance types across availability zones of Amazon Web Services. We have deployed this model on several Globus Galaxies instances. We present results in Section 5 that quantify the benefits of our approach.

2.6 Deployment and cloud hosting

A single instance of the Globus Galaxies platform requires the deployment of several different software packages, including Galaxy, Globus Connect Server, HTCondor, elastic provisioner, and NFS. A Globus Galaxies instance can be deployed in many configurations, depending on whether different components are deployed in the same or different cloud instances. A typical deployment bundles all management components (Galaxy, HTCondor master, NFS, GridFTP, elastic provisioner) on a single head node instance (m1.xlarge instance type). This instance is only used to run computationally limited tasks such as data transfers, a Galaxy Web server and a Galaxy database. All other computation is sent to an HTCondor queue for execution on worker nodes that are deployed when required. A typical instance of Globus Galaxies for genomics includes 2-8 TB of storage. When used for genomics, approximately 10 GB of this storage is used for tools, 55 GB for reference genome indices used by these tools, a small volume for shared Galaxy state such as scripts, and the remainder for user data.

Instances of the Globus Galaxies platform are operated as cloud-hosted SaaS. At present we manage over 30 different Globus Galaxies instances, 22 of which focus on genomics. As such, we require an easy to use and reproducible deployment method to quickly create new instances when required (i.e., when deploying new instances or when recovering from errors). To address these requirements we have developed a Chef-based system[14] capable of automatically deploying instances with all required software as well as contextualizing instances with specific settings. With this approach we can select the EC2 instance type and amount of total storage, and then deploy a full stack on a single instance or distributed across several instances, configure the elastic provisioner with appropriate policies, and add DNS entries for the Galaxy instance and Globus Connect Server used for data transfer.

3. Challenges using cloud resources

In contrast to HPC-backed scientific gateways, the Globus Galaxies platform is designed to be deployed on elastic cloud resources. This approach provides a platform on which analyses can be scaled and removes the need for researchers to own and operate their own resources. However, it also presents new challenges related to how instances are provisioned and used:

Resource selection: Cloud computing providers have many different instance types with varying resource capacity (e.g., EC2 has over 20 instance types). Many instance types also have different storage and networking configurations. Thus, selecting the “best” instance type for a given analysis is non-trivial. Moreover, the performance of different instance types is not necessarily proportional to cost, that is, a resource that costs twice

as much may not execute a workflow twice as fast. Some tools also require particular resources to be able to run while others might not be able to take advantage of advanced capabilities (e.g., multi-cores). Automated methods are required to accurately compare execution time (and cost) on different instance types. Such information can then be used by the elastic provisioner when determining what instance type to provision. As shown in Section 5 manual profiling approaches provide significant benefits; however this approach is time consuming.

Pricing models: Many cloud providers offer different pricing models, such as on-demand and spot prices. While spot pricing provides a way to obtain cheaper resources, instances may be terminated if the spot price exceeds the bid price. Therefore, careful scheduling and bidding policies must be used to avoid failure and cost increases. For example, when conducting long running analyses on demand instances may be used to alleviate risk, similarly, bid prices could be increased as an analysis executes to reflect the increasing cost associated with re-execution. Our cost-aware scheduling approach takes some of these considerations into account, however more sophisticated approaches may further improve efficiency and reduce cost.

Parallelization: Many scientific analyses are inherently parallelizable. While cloud providers offer enormous elasticity, it is challenging to deploy many instances, parallelize execution across these instances, and manage simultaneous execution efficiently. Moreover, parallelism may exist at many levels of a workflow. We currently support parallelization at the workflow and tool levels, allowing researchers to execute analyses over many input datasets at a single time and parallelizing components that are executed many times using swift wrappers. There is great potential to further improve this approach via automated introspection of workflows and tools, automated deployment of swift wrappers for loop-based execution, and exploiting parallel tools such as MapReduce-based variant callers.

Tool discovery, installation and execution: The Globus Galaxies platform provides a model for finding and executing tools, however to do so users must first find new analysis tools, develop wrappers, install them and their dependencies on different infrastructures or develop contextualization recipes to manage their deployment. Users that do not use Globus Galaxies must also manage execution on variable input datasets. Methods are needed to further simplify the process of finding, installing, and executing these tools in a cloud context.

Security: Research data is often sensitive and cannot be shared or accessed publicly. For example, biomedical research data may include potentially identifiable health information, such as faces or names, and other research data may contain important and valuable scientific data. It is therefore crucially important that such information is not publicly accessible. We have focused initially on analyzing anonymized and non-identifiable data and have recently developed a HIPAA-compliant Globus Galaxies deployment for genomics in partnership with REAN Cloud Solutions [15].

4. Globus Galaxies platform instances

Over the past three years we have created eight science gateways using the Globus Galaxies platform. Here we describe six of these implementations.

4.1 Genomics

Rapidly decreasing sequencing costs and the proliferation of gene sequencers have transformed biological research from being limited by data to being limited by computation. Increasingly,

researchers are processing hundreds of sequenced genomes as part of a single study to determine the statistical significance of variants. Analysis of large NGS datasets requires orchestrated workflows composed of many analysis tools and execution of these workflows using high performance computers and parallel algorithms to analyze data in a timely fashion. However, few research labs are equipped with specialized resources and personnel to conduct such analyses.

To address these requirements we created Globus Genomics, our first instance of the Globus Galaxies platform. We customized the deployment with over 1000 analysis tools—many of which perform similar tasks but have different strengths and are used by different user communities. Researchers studying neurodegenerative disorders, Tourette’s syndrome, diabetes, autism, breast cancer, molecular psychiatry, and cancer research have all used this platform in their everyday research activities. In these projects we have developed many workflows ranging from common variant calling, ChIP-Seq, RNA-seq, exome and whole genome analysis through to novel new workflows such as our consensus genotyper for exome sequencing variant caller [16]. Many workflows, often developed in collaboration with external researchers using the platform, have since been made available across the platform and have been used by others in different research groups and even research domains

Globus Genomics uses the complete Globus Galaxies platform, enabling execution of computationally intensive and long running genome analysis workflows. Given the huge number of tools included in this deployment we have developed a customized Globus Genomics AMI that includes several thousand commonly used genomics tools as well as the client software required to connect to the Globus Galaxies platform. We build upon CloudBioLinux [17] which includes a collection of leading NGS analysis tools; we regularly expand this set with additional tools identified via periodic user surveys. This AMI is used as the basis for creating worker nodes on which to execute tools.

Adoption of Globus Genomics by biomedical researchers has been rapid. We have created 22 Globus Genomics instances for different groups and more than 100 researchers, university research cores, and commercial users have used these instances to date.

4.2 Cardiovascular research

Researchers studying cardiac diseases conduct analysis of complex datasets composed of time-series, imaging and genomic data. As a rapidly developing research area there has been a significant growth in the number of analysis tools available and the types of analyses conducted. In particular, researchers are increasingly looking to integrate diverse data types and to increase sample sizes to conduct broad scientific studies. Thus, researchers are faced with challenges composing best practices workflows, finding and employing new research tools, and scaling analyses to include potentially large ECG and genomic data over increasingly large cohorts.

As part of the Cardiovascular Research Grid (CVRG) [18] we deployed the Globus Galaxies platform to support the analysis of cardiovascular research data. In this instance we have added support for a wide range of analysis algorithms, including Physionet algorithms [19] for ECG analysis, and various cardiac computational models including Calcium Spark, Hodgkin Huxley and Cardiac Myocyte models. We have developed a pre-built AMI that contains these tools, as well as many other leading cardiovascular, imaging, and genomic analysis tools. Using the standard Globus Galaxies platform model this AMI is used to

elastically provision worker nodes when required. All analyses are conducted on EC2 using the standard Globus Galaxies elastic provisioner. In collaboration with researchers at Johns Hopkins University we have added support for various ECG data types as well as integrating with WaveFormECG [20], a web-based tool for analyzing ECG data. This integration allows users to browse and select data visually via WaveFormECG and then run analyses using tools and workflows available in Galaxy.

The CVRG Globus Galaxies instance has been well adopted by the CVRG community. We are currently in the process of integrating Eureka! [21] a clinical data warehouse developed at Emory University. This integration will allow researchers to select a cohort of patients using Eureka! and invoke Globus Galaxies workflows that analyze the patients' genotypic data or ECG data

4.3 Climate and crop impact modeling

Understanding the impacts of climate change and the effect of various response strategies is an area of continued investigation. Researchers focus on integrating and applying many models and simulations to investigate the effect of various properties on different aspects. For example, analysis of the impact on crops may leverage different climate models, climate scenarios, economic models, and crop models at various spatial and temporal scales. The increasing size of data (e.g., satellite, climate models) combined with the complexity and number of models used necessitates the use of significant computational resources and also the orchestration of many different tools. In addition, to compute challenges climate researchers also face challenges related to finding data and transforming it into usable formats.

To address these challenges we have created the Framework to Advance Climate, Economic, and Impact Investigations with Information Technology (FACE-IT) [22] using the Globus Galaxies platform. FACE-IT provides access to a range of data, simulation models and analysis tools aimed at crop and climate impact assessment. Using FACE-IT users are able to experiment with different simulations, using various climate models and input datasets as input to various crop models. Researchers use this system to evaluate current conditions, future conditions with current approaches, and future conditions with adapted approaches. FACE-IT includes a number of new tools related to different models and also the translation of different data types such that they can be used by different tools.

In addition to adding many climate and crop modeling tools included in FACE-IT we have worked with earth-system-data providers to make common data sources available (e.g., Daymet daily surface weather dataset). We have also added support for new climate data types that support compressed and composite dataset management and type detection. For example, we have introduced a new NetCDF schema that is able to support NetCDF data, a common binary format used for earth-system data.

4.4 Medical imaging

Medical imaging, like other medical research domains, has fluctuating computational requirements based on when subjects are scanned and when new analysis routines and approaches are developed. The size of data generated by medical imaging researchers is steadily increasing with scanner resolution and new imaging techniques now leveraging multiple imaging gradients. Multi-subject cohorts routinely consume many gigabytes or even terabytes of storage. Image analysis workflows often build on computationally intensive algorithms such as segmentation, 3D rendering, and atlas registration.

In collaboration with researchers studying the diagnosis of Traumatic Brain Injury using Diffusion Tensor Imaging—an MRI technique capable of observing white matter changes in the brain—we have developed a DTI analysis gateway based on the Globus Galaxies platform [23]. These researchers use the gateway to develop new workflows that investigate the automatic extraction of clinical TBI biomarkers from DTI images. As the analysis of DTI for TBI is a new field, the researchers face challenges related to adapting existing image analysis techniques from other fields, integrating their own analysis routines, developing exploratory workflows, and frequently changing the analysis tools on which their workflows are based.

To meet these requirements we use the Globus Galaxies platform's support for dynamically contextualized instances. In this model, the researchers maintain a list of required software as well as the commands to install the software as a cloud-init script. The elastic provisioner uses this information to dynamically configure virtual machines initialized with required tools on demand. This approach allows the researchers to rapidly change the software that is used in their workflows and to have these changes instantly reflected when running analyses.

The researchers have used the DTI analysis gateway to create new biomarker extraction workflows and also to conduct research studies using these workflows. In particular, they have created several workflows for automatically extracting functional anisotropy measurements across a set of predefined regions of interest using a number of commonly available analysis tools. They have applied these workflows to study the reproducibility of automated measurements over a cohort consisting of over 100 scans.

4.5 Cosmology

Cosmology researchers are swamped with high-resolution datasets derived from large-scale sky surveys. From these surveys high-resolution simulations of cosmological structure formation have provided new insight into dark energy and dark matter. While huge advances have been possible, many researchers are limited by the huge data sizes and requirement for leadership class computing facilities when running simulations. The datasets used in these studies are often 100s of terabytes and sometimes even petabytes.

Building on the Globus Galaxies platform we have developed PDACS (Portal for Data Analysis Services for Cosmological Simulations) [24]. PDACS aims to increase the value of cosmological simulations by improving access to resulting datasets, providing access to advanced cosmological models, and providing the ability to access and execute tools on advanced computational resources.

Building upon the modular Globus Galaxies architecture, we have modified the system to support high performance cosmology workloads. In particular we have added a range of cosmology tools (Friends-of-Friends, halo finders, spherical over density, concentration measurements, etc.) and new data types for metadata propagation (including SQLite-based data types), and extended the execution model to support supercomputing facilities. Building upon Galaxy's runner interface we have added support for executing analyses with NEWT [25]—a REST API job submission interface used to submit jobs to computers at the NERSC supercomputer center. We have also extended our cloud-based execution model to use private cloud resources provided by Argonne National Laboratory's Magellan Cloud. As cosmology is inherently a visual field, we include a JavaScript interface for plotting graphs and support for interactive analysis capabilities.

4.6 Materials science

Computational materials science is a compute-intensive discipline in which simulations are used to model and investigate complex materials properties. Workflows are used to understand molecular dynamics, electronic structure, and conductance of materials, often with a feedback loop to compare results against experimental findings. Materials simulations typically consume many hours of compute time across many different analysis tools and often involve many gigabytes of raw and derived data.

One example of a materials analysis workflow is the investigation of the electrical conductance of layered oxides that strongly depends on the atomic structure of the interface between materials. Molecular dynamics (MD) calculation can quickly (minutes to hours using LAMMPS) determine the typical structure of the interface. Conductance calculations involve a parameter sweep through multiple electronic structure calculations taking days on leadership-class resources (using Siesta/Sméagol libraries). MD interatomic force fields need to be fitted to experimental data using complex multistep machine learning calculations. Finally data movement and conversion is required to assemble the steps into a complete workflow.

In collaboration with materials researchers at Argonne National Laboratory we have developed eMatter—a materials science gateway that makes it easy to access data, simulation tools, and to combine individual simulations into multi-scale workflows such as the one described above. eMatter is configured to use on-demand, spot cloud resources to conduct analyses, however, given the large-scale nature of these simulations we will soon leverage leadership class computing facilities at Argonne National Laboratory. We are also investigating making it possible for researchers to dynamically and easily upload their own tools as modules that will enable other users to access data and run new simulations.

5. Evaluation

Our evaluation focuses on three aspects: 1) production usage of the Globus Galaxies platform over the past three months. 2) Use of cloud computing provisioning approaches to scale analyses, 3) cost-aware elastic cloud provisioning.

5.1 Production resource usage

Figure 2 shows the cumulative resource usage in total compute hours from seven production Globus Galaxies deployments over the previous three-month period. It is important to note that all Globus Galaxies instances are configured to allow multiple jobs to be scheduled to the same worker node if the number of cores required by a task are available. The instances are arbitrarily numbered for discussion purposes.

The figure highlights the variability of usage requirements for all of the deployments. For example, instance 7 shows relatively consistent usage over the period with only small variations on a weekly basis up until week 14. Instance 4 has considerable usage between weeks 8 and 11, but little usage in other periods. In total, over 25,000 compute hours have been used by all instances in this three-month period. In most cases the instances provisioned contain eight CPUs meaning that up to 200,000 CPU hours have been consumed in three months. These usage patterns highlight the value of using on-demand computing resources as instances dynamically provision additional resources only when required.

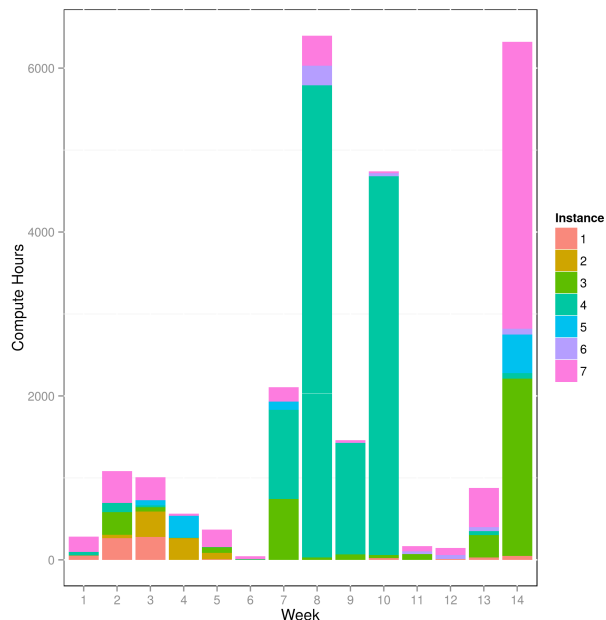


Figure 2: Total compute hours used by seven Globus Galaxies platform deployments.

5.2 Production tool usage

The various Globus Galaxies deployments collectively have thousands of tools available to users. Table 1 presents the 20 tools that have consumed the most compute hours over a three-month period. The table shows the frequency of tool execution, average execution time for the tool, and the percentage of total execution time consumed by the tool. The top 10 tools account for 37.4% of all executions and 81.9% of the execution time. The top 20 tools account for 64% of all executions and 92.8% of the remaining execution time. These results are heavily skewed towards genome analysis as the majority of Globus Galaxies usage to date has been via Globus Genomics. Knowledge of such information can be used to customize Globus Galaxies deployments to optimize execution for frequently used tools.

Tool	Freq.	Avg Time	Time (%)
fastq_groomer_parallel	1047	5:48:25	51.33
sickle	454	1:57:09	7.48
bwa_mem	469	1:07:55	4.48
CONVERTER_sam_to_bam	52	7:39:45	3.36
cru3d_simulation	11	36:08:25	3.36
gatk2_haplotype_caller	26	13:01:38	2.86
rgPicardMarkDups	561	0:36:11	2.86
bowtie_wrapper	60	4:19:39	2.19
Filter1	135	1:51:54	2.13
Fastqc	508	0:25:04	1.79
isaac_align	50	3:46:27	1.59
samtools_flagstat	483	0:21:31	1.46
tophat2	21	8:07:40	1.44
picard_BuildBamIndex	394	0:23:12	1.29
picard_ARRG	462	0:16:26	1.07
csht_fastq_quality_filter	165	0:40:35	0.94

picard_SortSam	421	0:15:26	0.91
samtools_sort	112	0:52:09	0.82
trim_galore	182	0:29:14	0.75
gatk_depth_of_coverage	79	1:02:28	0.69

Table 1: Top 20 tools ordered by total compute time consumed.

5.3 Provisioning approaches

The Globus Galaxies platform supports two models for elastically provisioning cloud instances: 1) using a preconfigured AMI; or 2) dynamically contextualizing instances when required. Figure 3 shows the time taken to run an example DTI imaging workflow[23] when using these two approaches. The graph also includes a baseline approach when reusing an existing worker node that has already been provisioned.

Using a preconfigured AMI takes approximately eight minutes longer than reusing an existing instance, whereas dynamic contextualization takes approximately 15 minutes longer. In both cases, much of this time is spent waiting in the HTCondor queue before an instance is provisioned and for the spot instance to be started. The difference between dynamic contextualization and using a preconfigured AMI is approximately four minutes. This time is spent downloading required packages and installing and configuring these packages. In this workflow, only six software packages are installed dynamically, a small number compared with the thousands of tools available in gateways such as Globus Genomics. While the overhead of dynamic contextualization may be acceptable for longer running workflows with few tool dependencies, it represents a significant overhead on shorter duration workflows with many dependencies. Others have investigated startup time of different instance types, with different operating systems, using different provisioning models, in different regions and at different times of the day, with similar results [26]. Such information can also be used to improve scheduling performance, favoring reuse and incremental contextualization over new deployments.

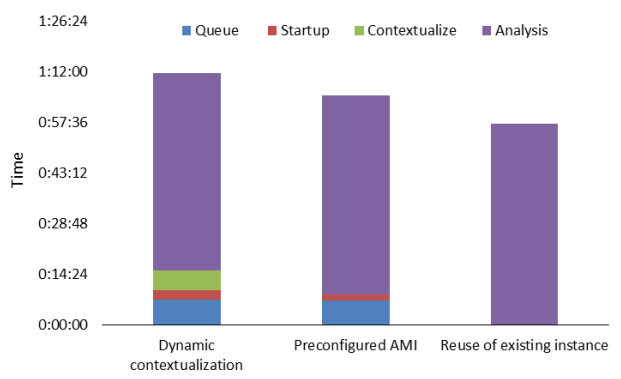


Figure 3: Total workflow execution time when using different provisioning approaches for an example DTI workflow.

5.4 Cost aware scheduling

Given the huge range of cloud instances available and the different pricing models used by commercial cloud providers there is potential for significant cost savings if appropriate profiling and scheduling approaches are employed. Figure 4 shows the execution time and cost for an example DTI workflow on various EC2 instance types, with both Elastic Block Store

(EBS) and instance store storage. The cost is estimated based on the on-demand price as well as the observed low and high spot price over a three-month period. The figure shows that while execution time may be fastest using the m3.2xlarge instance type, the total cost is greater than several other instance types. It also shows the increased volatility of spot prices may result in much lower or much higher costs depending on when the instance is provisioned. Using these results we have configured the DTI gateway elastic provisioner to map these workflows to the most suitable instance types. We can execute the entire workflow in less than 40 minutes and for less than US\$0.05 (low spot) or US\$0.40 (on-demand) per subject.

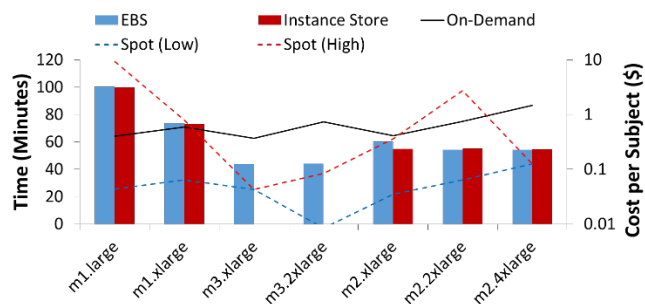


Figure 4: Execution time and cost for an example DTI workflow on various EC2 instance types.

As many of the tools used in this DTI workflow do not take advantage of multiple cores or large amounts of memory we can further improve cost efficiency by scheduling multiple workflows on the same instance. To do so efficiently, we have profiled the requirements of the tools used in the workflow and configure the elastic provisioner with an attribute that defines how many workflow instances can be scheduled to the same cloud instance. Figure 5 shows the reduced cost per subject obtained by scheduling several workflows (up to eight for m2.4xlarge) per instance. While significant cost reductions can be achieved, it assumes a number of workflows are executed concurrently and therefore is only enabled during large cohort analyses.

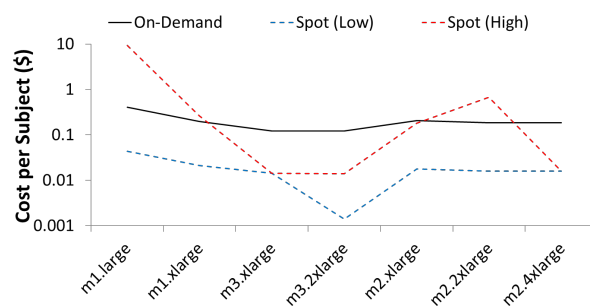


Figure 5: Execution time and cost for an example DTI workflow on various EC2 instance types.

We have conducted similar analysis of other workflows in other Globus Galaxies deployments to optimize execution. We are currently investigating methods to automatically select instances and concurrent execution based on workflow and instance profiling.

5.5 Application of cost aware scheduling

Figure 6 shows the cumulative cost of real workloads from seven production Globus Galaxies deployments. The solid lines show the cumulative cost using a simple scheduling strategy that uses

only a single instance type (m2.4xlarge) and a fixed bid price (\$2.00). The dashed lines show the cumulative cost when applying a flexible scheduling approach in which the scheduler adjusts instance types based on real-time spot prices across availability zones. In this case we surveyed the workflows used by these seven instances and identified five instance types that can satisfy the requirements of these workflows: m2.4xlarge, c3.2xlarge, cx.4xlarge, r3.2xlarge, and r3.4xlarge.

The results highlight the potentially significant cost savings when applying, even relatively simple, cost-aware scheduling strategies. In this case the total cost over a three-month period is reduced by up to 93% or over \$29,000. In each of the instances considered the average saving is 92.6%, with a minimum improvement of 84.1% and a maximum improvement of 97.4%.

6. Related Work

The Galaxy team operates a public Galaxy instance (usegalaxy.org) hosted at the Texas Advanced Computing Center. This free service is used by thousands of researchers to perform hundreds of thousands of analyses each month, subject however to some limits on data transfer and computer usage. Thus, users are able to experiment with Galaxy, for biomedical analyses, and are expected to deploy their own instances locally when they outgrow this service. The success of the public Galaxy service has been a motivation for the development of the Globus Galaxies platform to provide the same SaaS benefits while removing usage limitations and employing a pay per usage model.

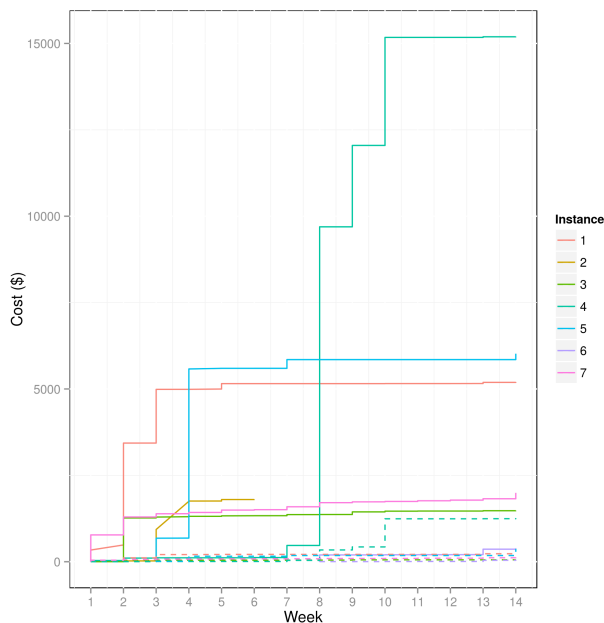


Figure 6: Total execution cost when using two different scheduling approaches.

The Galaxy team also created CloudMan [27]—a script-based approach for deploying Galaxy servers in the cloud. However, unlike the Globus Galaxies platform, which enables the creation of gateways that are operated as a Service for each user community, CloudMan provides only the functionality to make it easier to deploy a Galaxy cluster on the cloud. It still requires that users know and understand the operating complexities of cloud computing infrastructure; requires that they manage their own cloud account, handle the authentication and use of cloud APIs

and provision, in advance, the cluster used to execute analyses; and does not support elastic provisioning, high performance data transfer, or other features provided by the Globus Galaxies platform.

There are several SaaS and PaaS providers that focus on providing scientific services. These services have made transformative changes to everyday research activities, enabling researchers and scientific software developers to outsource important IT functionality to professionally hosted providers. One such example is Globus—a SaaS provider of research data management functionality. Globus allows researchers to transfer, synchronize and share large research datasets securely and efficiently via an intuitive web interface. Globus also exposes REST APIs allowing its features to be exploited as a platform. Through this model developers can outsource identity, group and profile management, user authentication, and data transfer, synchronization and sharing. Agave is another scientific service provider that supports authentication, authorization, job submission and data management. Rather than provide a usable web interface, these collective capabilities can be leveraged by scientific gateways to outsource functionality through a professionally hosted REST API. Globus Galaxies is differentiated by its focus on providing complete end-to-end SaaS solutions and builds upon these existing PaaS approaches to outsource some functionality.

Service-oriented approaches have long been applied to job submission systems. Services provide a standardized interface to access complex job scheduling capabilities and have been used both to abstract differences between schedulers and to provide advanced management functionality. Examples include web services GRAM[28], NEWT and Grisú [29]. However, like PaaS approaches these submission services provide only the building blocks on which more advanced functionality can be constructed.

HUBzero and Apache Airavata provide software frameworks to deploy science gateways—which can be deployed on the cloud. HUBzero is a platform for creating dynamic collaborative science “hubs” and is now used by over 40 groups across a range of domains. It is based on an extensible platform that allows execution of interactive simulation and modeling tools via a web browser. Like Galaxy, HUBzero is predominantly deployed by large user groups on their own resources; however the HUBzero team also provides a chef-based approach for deploying HUBzero on Amazon EC2 (similar in nature to CloudMan). Apache Airavata is a framework for creating science gateways and running computational workflows on distributed computing resources. Airavata provides a number of services such as job submission, workflow management, and application management that can be used to create web-based science gateways. Both HUBzero and Airavata can be used to create individual gateways, however the major difference between them and the Globus Galaxies platform is that they require users to modify, deploy and manage the gateway codebase rather than accessing these capabilities via a hosted SaaS model.

7. Summary

The Globus Galaxies platform provides a cloud-hosted approach for deploying SaaS science gateways. By integrating existing service providers for identity and transfer management, leveraging a leading workflow system, and elastically scaling analyses using on demand cloud instances the platform is able to meet the needs of a wide range of scientific user communities. Over the past several years we have applied the platform with a great deal of

success to a number of different domains. This experience has reinforced our belief that such an easy to use, flexible, and scalable approach is suitable for a wide range of users. The ability to leverage pay as you go resources means that users of the platform are able to avoid up front expenses for purchasing resources and therefore makes the platform valuable to resource limited laboratories.

Our focus on cost-aware scheduling has made it possible to optimize execution efficiency whilst also minimizing cost. Workflows executed by the platform are able to take advantage of low cost spot instances where required, respond to fluctuating instance costs, and even concurrently execute on shared instances in an effort to further decrease cost. These features in particular make the Globus Galaxies platform an ideal platform for creating scientific gateways for scientific users who have fluctuating resource requirements. It also provides a model for lowering operational costs and presents a path towards long term gateway sustainability as users pay for the resources used. Globus Galaxies allows us to rapidly develop and deploy new functionality while making the capabilities highly available as we leverage best practices of hosting, running analysis on cloud computing resources. Collectively, these features can lower time to science for a wide range of scientific users.

Acknowledgments

This work was supported in part by the U.S. Department of Energy, Office of Science, under contract DE-AC02-06CH11357 and the NIH through U24 GM104203 Bio-Informatics Research Network Coordinating Center (BIRN-CC) and R24HL085343 Cardiovascular Research Grid (CVRG). We would like to acknowledge generous research credits provided by Amazon We Services that helped support our work. We also appreciate the support of the Globus and Galaxy teams.

References

1. Foster, I., *Service-Oriented Science*. Science, 2005. **308**(5723): p. 814-817.
2. Rion Dooley, M.V., Dan Stanzione, Steve Terry, Edwin Skidmore. *Software-as-a-Service: The iPlant Foundation API*. in *5th IEEE Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS)*. 2012.
3. Foster, I., *Globus Online: Accelerating and democratizing science through cloud-based services*. IEEE Internet Computing, 2011(May/June): p. 70-73.
4. McLennan, M.K., R. Kennell, *HUBzero: A Platform for Dissemination and Collaboration in Computational Science and Engineering*. Computing in Science & Engineering, 2010. **12**(2): p. 48-53.
5. Marru, S., et al., *Apache airavata: a framework for distributed applications and computational workflows*, in *Proceedings of the 2011 ACM workshop on Gateway computing environments*. 2011, ACM: Seattle, Washington, USA. p. 21-28.
6. David Lifka, Ian Foster, Susan Mehringer, Manish Parashar, Paul Redfern, Craig Stewart, Steve Tuecke, *XSEDE Cloud Survey Report*.
7. Madduri, R.K., et al., *Experiences Building Globus Genomics: a Next-Generation Sequencing Analysis Service using Galaxy, Globus Online and Amazon Web Services*. Concurrency - Practice and Experience, 2013..
8. Goecks, J., A. Nekrutenko, and J. Taylor, *Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences*. Genome Biol, 2010. **11**(8): p. R86.
9. Allen, B., et al., *Software as a service for data scientists*. Commun. ACM, 2012. **55**(2): p. 81-88.
10. Ananthakrishnan, R., et al. *Globus Nexus: An identity, profile, and group management platform for science gateways and other collaborative science applications*. in *Science Gateway Institute Workshop, co-located with IEEE Cluster*. 2013.
11. Wilde, M., et al., *Parallel Scripting for Applications at the Petascale and Beyond*. IEEE Computer, 2009. **42**(11): p. 50-60.
12. Litzkow, M.J., M. Livny, and M.W. Mutka, *Condor - A Hunter of Idle Workstations*, in *8th International Conference on Distributed Computing Systems*. 1988. p. 104-111.
13. Amazon. *Amazon Web Services*.; Available from: <http://www.aws.amazon.com> [Accessed 21 January 2015].
14. Bo, L., et al. *Deploying Bioinformatics Workflows on Clouds with Galaxy and Globus Provision*. in *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion*.: 2012.
15. Hupp, S., *REAN Cloud Partners with Globus Genomics to Scale Genomics Research with HIPAA Compliance*: Available from: <http://tickerreport.com/press-releases/384283/rean-cloud-partners-with-globus-genomics-to-scale-genomics-research-with-hipaa-compliance/> [Accessed 21 January 2015]
16. Trubetskoy, V., et al., *Consensus Genotyper for Exome Sequencing (CGES): improving the quality of exome variant genotypes*. Bioinformatics, 2015. **31**(2): p. 187-93.
17. Krampis, K., et al., *Cloud BioLinux: pre-configured and on-demand bioinformatics computing for the genomics community*. BMC Bioinformatics, 2012. **13**: p. 42.
18. Winslow, R.L., et al., *The CardioVascular Research Grid Project*. AMIA Summits Transl Sci Proc, 2011.
19. Goldberger, A.L., et al., *Physiobank, physiokit, and physionet components of a new research resource for complex physiologic signals*. Am Heart Assoc. **101**(23): p. e215-e220 0009-7322.
20. Granite, S. *Waveform ECG*. 2014; Available from: <http://civrgrid.org/tools/waveform-ecg> [Accessed 21 January 2015]
21. Post, A.R., et al., *Temporal Abstraction-based Clinical Phenotyping with Eureka!* AMIA Annual Symposium Proceedings, 2013. **2013**: p. 1160-1169.
22. Montella, R.B., et al., *FACE-IT: a science gateway for food security research*. in *9th Gateway Computing Environments Workshop*. 2014. New Orleans.
23. Chard, K., et al., *A cloud-based image analysis gateway for traumatic brain injury research*, in *Proceedings of the 9th Gateway Computing Environments Workshop*. 2014, IEEE Press: New Orleans, Louisiana. p. 13-16.
24. Chard, R., et al., *PDACS: a portal for data analysis services for cosmological simulations*, in *Proceedings of the 9th Gateway Computing Environments Workshop*. 2014, IEEE Press: New Orleans, Louisiana. p. 30-33.
25. Cholia, S., D. Skinner, and J. Boverhof. *NEWT: A RESTful service for building High Performance Computing web applications*. in *Gateway Computing Environments Workshop (GCE)*. 2010, IEEE Press: p. 1-11
26. Ming, M. and M. Humphrey. *A Performance Study on the VM Startup Time in Proceedings of the 5th IEEE International Conference on Cloud Computing (CLOUD)*, 2012.
27. Afgan, E., et al., *Harnessing cloud computing with Galaxy Cloud*. Nat Biotech, 2011. **29**(11): p. 972-974.
28. Czajkowski, K., et al., *A resource management architecture for metacomputing systems*, in *Job Scheduling Strategies for Parallel Processing*. 1998, Springer. p. 62-82.

29. Altinay, C., et al. *High-Performance Scientific Computing for the Masses: Developing Secure Grid Portals for Scientific Workflows in Proceedings of the 6th IEEE International Conference on e-Science (e-Science)*. 2010.

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.